Al-Mustaqbal University College

1

Department of Advanced digital design

Class: four

Subject: Medical electronic system lab

Lecturer: Dr. zahraa hashim

Eng. Ali Ibrahim

Eng. Hussein Mazin

# Arduino

# is used in many educational programs around the world

# Arduino Types:

# 1-Arduino UNO



1

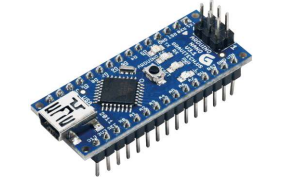
2

# 



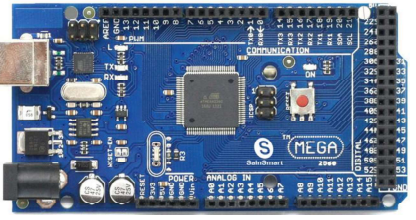
**Fig(1)**

2- Arduino Nano



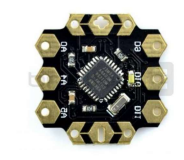
**Fig(2)**

3-Arduino Mega 2560



**Fig(3)**

4- Cheapduino



**Fig(4)**

Here is a sketch that demonstrates the example values shown in Table of Bit operators :

//\*

\* bits sketch

\* demonstrates bitwise operators

\*//

void setup() {

Serial.begin(9600);

}

void loop(){

Serial.print("3 & 1 equals "); // bitwise And 3 and 1

Serial.print(3 & 1); // print the result

Serial.print(" decimal, or in binary: ");

Serial.println(3 & 1 , BIN); // print the binary representation of the result

Serial.print("3 | 1 equals "); // bitwise Or 3 and 1

Serial.print(3 | 1 );

Serial.print(" decimal, or in binary: ");

Serial.println(3 | 1 , BIN); // print the binary representation of the result

Serial.print("3 ^ 1 equals "); // bitwise exclusive or 3 and 1

Serial.print(3 ^ 1);

Serial.print(" decimal, or in binary: ");

Serial.println(3 ^ 1 , BIN); // print the binary representation of the result

byte byteVal = 1;

int intVal = 1;

byteVal = ~byteVal; // do the bitwise negate

intVal = ~intVal;

Serial.print("~byteVal (1) equals "); // bitwise negate an 8 bit value

Serial.println(byteVal, BIN); // print the binary representation of the result

Serial.print("~intVal (1) equals "); // bitwise negate a 16 bit value

Serial.println(intVal, BIN); // print the binary representation of the result

delay(10000);

}

This is what is displayed on the Serial Monitor:

3 & 1 equals 1 decimal, or in binary: 1

3 | 1 equals 3 decimal, or in binary: 11

3 ^ 1 equals 2 decimal, or in binary: 10

~byteVal (1) equals 11111110

~intVal (1) equals 11111111111111111111111111111110

Bitwise operators are used to set or test bits. When you And or Or two values, the operator works on each individual bit. It is easier to see how this works by looking at the binary representation of the values. Decimal 3 is binary 00000011, and decimal 1 is 00000001. Bitwise And operates on each bit. The rightmost bits are both 1, so the result of And-ing these is 1. Moving to the left, the next bits are 1 and 0; And-ing these results in 0. All the remaining bits are 0, so the bitwise result of these will be 0. In other words, for each bit position where there is a 1 in both places, the result will have a 1; otherwise, it will have a 0. So, 11 & 01 equals

Compound bitwise AND (&=)

x &= y; // equivalent to x = x & y;

Parameters

x: a char, int or long variable

y: an integer constant or char, int, or long

Example :

First, a review of the Bitwise AND (&) operator

0 0 1 1 operand1

0 1 0 1 operand2

----------

0 0 0 1 (operand1 & operand2) - returned result

Compound bitwise OR (|=)

x |= y; // equivalent to x = x | y;

Parameters

x: a char, int or long variable

y: an integer constant or char, int, or long

Example :

First, a review of the Bitwise OR (|) operator

0 0 1 1 operand1

0 1 0 1 operand2

----------

0 1 1 1 (operand1 | operand2) - returned result

Discussion:

1-What is Arduino and what is it used for?

2-What is the programming language used with Arduino?